Patent Pending Related to GOBS Technology. Authorization for Use/Disclosure Required.

Carderock Division
Naval Surface Warfare Center
Bethesda, Maryland 20817-5700

NSWCCD-26-TR-1998/xx October 1998

Total Ship Systems Directorate
Research and Development Report

Leading Edge Advanced Prototyping

For Ships (LEAPS):

Geometry Object Structure (GOBS)

Version 2.01

by
Robert Ames and Richard T. Van Eseltine

TABLE OF CONTENTS

TABLE OF FIGURES

GEOMETRY OBJECT STRUCTURE – LEAPS/GOBS

Background

The LEAPS Geometry Object Structure, LEAPS/GOBS, is a new model for the construction, representation, and interrogation of complex geometric models. It is important to understand the background information presented because of some the unique entities of GOBS modeling. An overview of idealized geometry is presented followed by a description of the classes available for construction. More detailed information on these classes is available in the *Leaps Reference Manual.*

The GOBS model purports that geometric product model data is defined and represented as ‘views’ of geometric objects. The word “view” is in quotes because, as will be explained later, is actually an object that appears as geometry. This is not to say that GOBS does not allow geometric objects to represent geometric product model data only that another more powerful approach is available. This is contrary to most representations where the geometry defines the view and the object simultaneously.

To understand one aspect of this new geometry topology, an example of three compartments within a ship, depicted in Figure 1, will be used to for demonstration purposes.

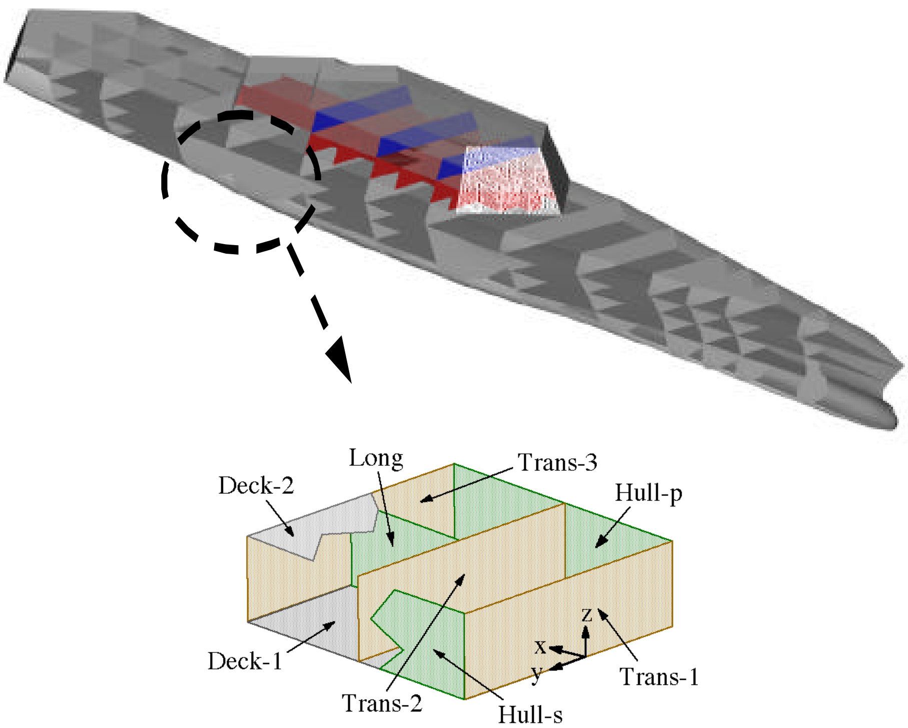

Figure 1 - Three Compartment Test Case

This three compartment case, while simplistic in appearance, actually poses a number of challenges to product modeling. Consider the "knowledge" that must exist at transverse bulkhead 2, (Trans-2). This bulkhead plays a number of roles one of which is the boundary of three compartments. The bulkhead is connected to the hull, port and starboard, the longitudinal, and both decks above and below. In addition, there are locations on this bulkhead that may be of interest to analysts such as the corner points at intersections with other surfaces (longitudinal, hull, deck, etc.). This bulkhead also plays a role as a boundary, or *Face*, of each individual compartment. These boundaries can be described as "views" of the bulkhead as seen by each compartment and unique to each compartment. Consider also that the object Trans-2 may play a role, or roles, in many other "views" such as a watertight bulkhead bounding a zone on the ship.

Product Model Views

In GOBS "views" of product model data are actually objects that compose existing geometry into unique physical objects. Similarly there are views that associate physical objects into like groupings. Views that create physical objects from geometry elements are called *Topological Views.* Views that associate *Topological Views* into common groups are called *Common Views*.

Topological Views

The term *Topological View* is foreign to most familiar with geometric modeling. Its best to think of them as tradition surfaces, trimmed surfaces, and Brep solids, with additional capability. The construction of *Topological Views* allows for member objects, like surfaces and solids, used in the creation of a *Topological View,* to also play a role as geometric members in others *Topological Views*. To understand *Topological Views,* an understanding of all of the GOBS classes is necessary. These classes are described later in this document in the semantic of computational geometry. For the uninitiated, an explanation of surfaces, trimmed surfaces or faces, solids, and other terms associated with computational geometry can be found in at http://ocean.dt.navy.mil/. We'll assume from this point forward that these concepts are understood.

Common Views

Common Views do not have any spatial constraints, unlike *Topological Views*, they are simply a logical grouping of *Topological Views. Common Views* can also have other *Common Views* as members. *Common Views* are the primary vehicle by which domain analyst or designers will view or interrogate the product model. One example of of a *Common View* could be "Habitability Spaces on Deck 3". Another *Common View* called "Ship Habitability Spaces" could contain the *Common View* "Habitability Spaces on Deck 3" as a member. Similar uses of *Common Views* could include "Exterior Surfaces", "Compartments", "Machinery Spaces", or "Mast".

Shape Objects

Some distinctions should be made of the differences between GOBS shape objects and what can be considered typical geometric entities in applications that use and compose geometry such as CAD systems. In GOBS, geometry (*Topological Views*) is the association of shape and *Properties*. Current shape objects are *Surfaces*, trimmed surfaces (*Faces*), and manifold brep *Solids*.

One major difference in GOBS modeling is the representation of *Faces*. Currently CAD systems today consider a *Face* to be composed of a single *Surface* bounded by a single outer boundary and any number of inner boundaries. The typical CAD model does not allow the underlying *Surface* to be used in the construction of any other *Face.* It requires that a copy of that *Surface* be made. GOBS, on the other hand, allows for a single *Surface* to be used in the construction of any number of *Faces*; where the *Face* object contains reference to one *Surface*, one outer *EdgeLoop*, and any number of inner *EdgeLoops*. This concept is illustrated more clearly in Figure 2 where the deck on a ship is shown highlighting three *Faces* used as compartment boundaries. All three *Faces* share a common deck *Surface* and are defined by a selection of *Edges* that compose a bounded *EdgeLoop.*

Because *Surfaces, Faces*, and *Solids* are shape objects they have no *Properties*. In GOBS the *Topological View* class associates member shape objects with physical characteristics or *Properties* and can be thought of as a geometric component, or part. The *Topological View* has *Properties* of a physical or performance nature, where the underlying *Surface, Face*, or *Solid* object, is simply providing information on its shape. As *Topological Views* are composed, the grouping into *Common Views* is the next natural step.

In Figure 2, *Topological Views* of regions on a deck are illustrated. In this case they appear as "Comp 1 Deck", "Comp 2 Deck", and "Comp 3 Deck". Each *Topological View* use *Faces* ("FA1", "FA2", "FA3") to define their shape within compartments (*Common Views* named "Compartment 1", "Compartment 2", "Compartment 3"). Similarly these *Common View* compartments are also shown as members of a single *Common View* defining a zone ("Zone 1"). In summary, this example demonstrates how space with each compartment derives their shape from *Faces*. These *Faces* are defined as simple boundaries (*EdgeLoops)* on single underlying geometric element which is a *Surface* (Deck 1) defining the entire deck shape. These spaces are then associated in a *Common View* to support design domain knowledge.

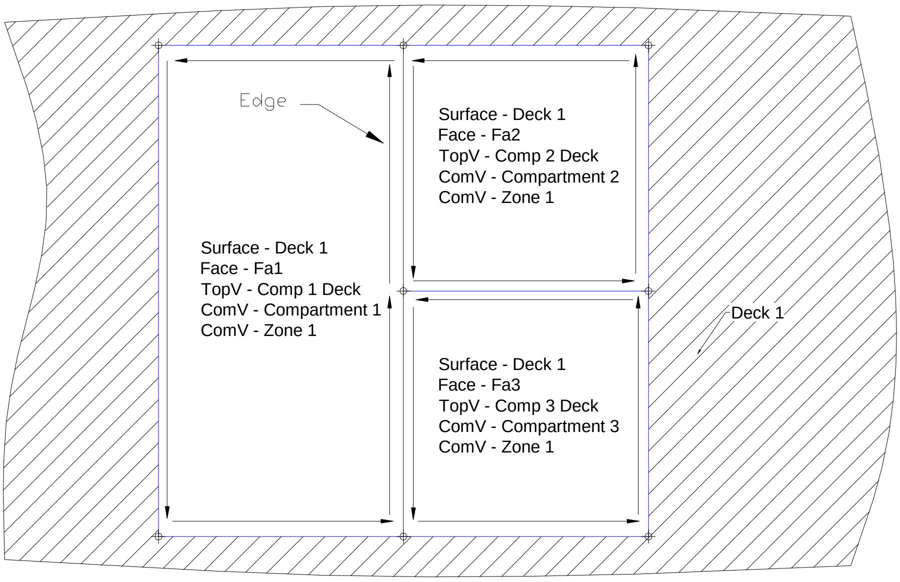

Figure 2 - Three Compartments on Deck 1

Figure 3 further illustrates this point using a bulkhead of a carrier containing three compartment *Topological Views (Faces*). These *Topological Views* could participate in multiple *Common Views*. Should the need arise, a *Common View* could be composed and identified as “Habitability Space Bounded by Bulkhead 27”. Likewise, individual *Topological Views* may also be in other *Common Views.* One logical grouping would be the grouping of *Topological Views* that compose a compartment or zone. In this scenario there might be *Common Views* called “Compartment 51- Crew Berthing”, “Compartment 52– Crew Mess”, and “Compartment 38 – Officers Quarters”, etc.

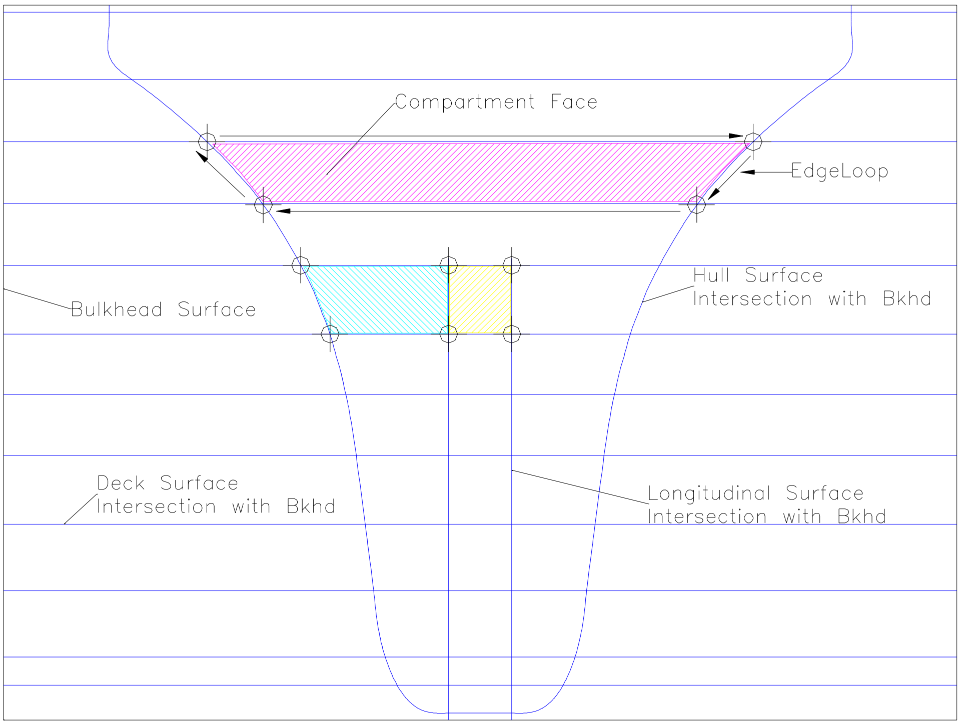

Figure 3 - Carrier Bulkhead with Three Compartment Topological Views (Faces)

Topological Views are one of two main concepts critical to the GOBS methodology that is important to comprehend. The other is geometric connection. In GOBS, there are two objects that provide a logical and explicit connection between shape construction entities. These objects are called *CoEdge* and *CoPoint*. GOBS uses these elements to allow for the connection of entities, such as edges, and points, through geometric reference.

To better understand how these construction entities are used to compose geometry it is necessary to understand some of the other GOBS classes and their associations. The next section will go through each GOBS class and describe their role in building a robust GOBS *Structure*.

GOBS Classes

GOBS is composed of a number of objects that when combined provide for a powerful physical product model metaphor. These classes are defined below. GOBS is not, however, limited to these objects nor their analytic representation.

Structure

A *Structure* object creates and manages all Gobs objects that geometrically describe the design object. A *Structure* can be viewed as a collection of objects or entities that compose a part or structure. This can be anything from a ship to a helicopter. A *Structure* should be seen as largest representation of the product or part.

Common View

A *Common View* object groups *Topological View* objects and/or other *Common View* objects into a logical view of a *Structure* object. Objects may be members of multiple *Common Views*.

Topological View

A *Topological View* object is physical object used in the construction of a *Structure*. It defines physical objects as having shape (*Surface*, *Face* or *Solid* object) and physical or behavioral properties (*PropertyGroup* and *Property* objects).

Surface

A *Surface* object is a non-uniform rational b-spline representation of a surface in Cartesian space.

Pcurve

A *Pcurve* object is a parametric spline curve on a *Surface* object. Its range is the domain of the *Surface* it is mapped to.

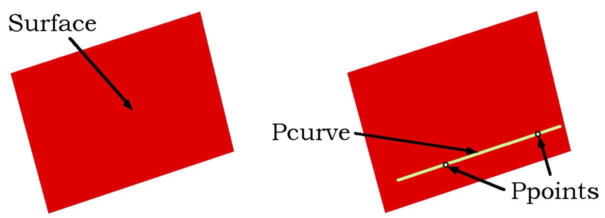

Figure 4 - Surface, Pcurve and Ppoint Object

Ppoint

A *Ppoint* object is a parametric point on a *Pcurve* object.

CoPoint

A *CoPoint* object is a collection of *Ppoint* objects that a the same point in model space.

Edge

A *Edge* object is a segment of a *Pcurve* object defined by two *Ppoints*. Each *Edge* is associated and EdgeOpposite (created automatically at construction) which is an *Edge* of opposite orientation.

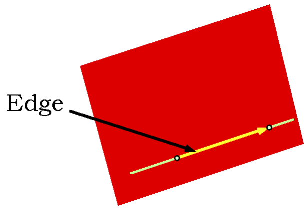

Figure 5 - Edge Object

EdgeLoop

An *EdgeLoop* object contains a series of connected *Edge* objects that form a closed loop. Because each *Edge* has and opposite each *EdgeLoop* has an opposite with reverse orientation. An opposite *EdgeLoop* is created at each *EdgeLoop* construction

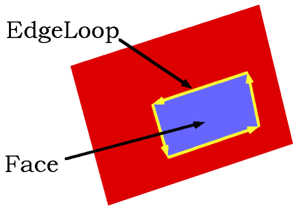

Figure 6 - EdgeLoop

Face

A *Face* object contains one or more *EdgeLoops* of which there is one outer *EdgeLoop* and zero or more inner *EdgeLoops*. A *Surface* object may participate in any number of *Face* objects. Like the *Edge* and *EdgeLoop*, each *Face* has an opposite whose normal points in the other direction.

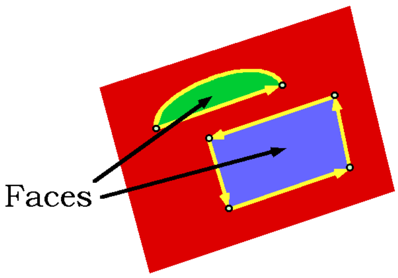

Figure 7 - Face Object

CoEdge

A *CoEdge* object is collection of *Edge* objects that represent the same model space edge. CoEdges cannot have any interior Ppoints or CoPoints, See Figure 8.

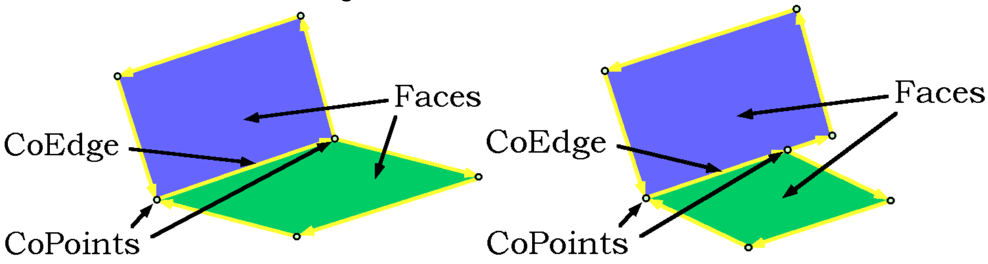

Figure 8 - CoEdge Object

A *CoEdge* contains a single n-dimensional spline function that contains the domain space of each member *Edge* as range variables of its function. Further information on *CoEdges* can be found in APPENDIX .

Oriented Closed Shell

An *Oriented Closed Shell* object is a collection of *Face* objects that form a closed shell.

Solid

A *Solid* object contains one or more *Oriented Closed Shells* of which there is one outer *Oriented Closed Shell* and zero or more inner *Oriented Closed Shell*. An Oriented Closed Shell object may participate in any number of Solid objects as shell boundaries.

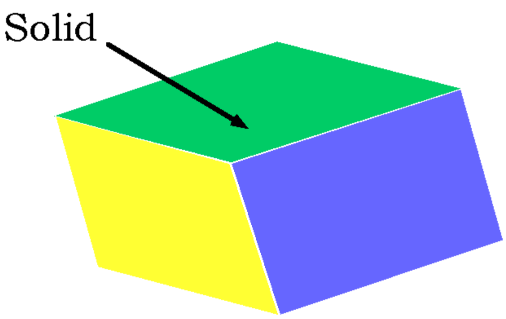

Figure 9 - Solid Object

APPENDIX A

<u>CoEdges</u>

A fundamental feature of LEAPS/GOBS is the *CoEdge*. A *CoEdge* is a connection object that contains information about the connection of *Edges* that are coincident in Cartesian space but not defined in Cartesian space. Not only does a *CoEdge* define *Edges* as having the same physical location, it also relates the parametric space to the physical space along the *CoEdge* for all member *Edges*. There are many applications that can leverage this knowledge. One identified application is analysis preparation. The *CoEdge* provides a unique capability for the discretization of the geometry for analysis. Put another way, the *CoEdge* contains information necessary to compose idealized geometry for analysis, evaluation, or post processing. Essentially, a *CoEdge* knows all *Edges* located on each *Surface* and declares them to be equivalent in 3 space, see Figure 5. It also contains an n-dimensional spline function which maps the parameterization of each edge into a single function. This allows for the continuity of points along one *Surface* to migrate to another *Surface* without having to perform closest point approximations.

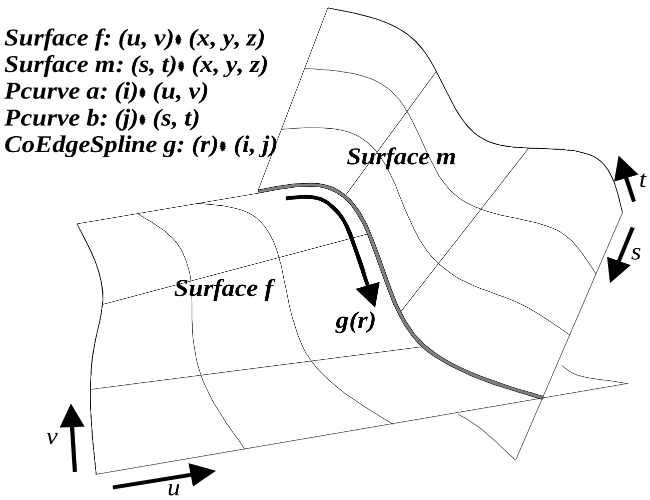

Figure 10 - CoEdge Spline Dependencies

The parametric association for the *CoEdge* and its related objects is illustrated in Figure 11.

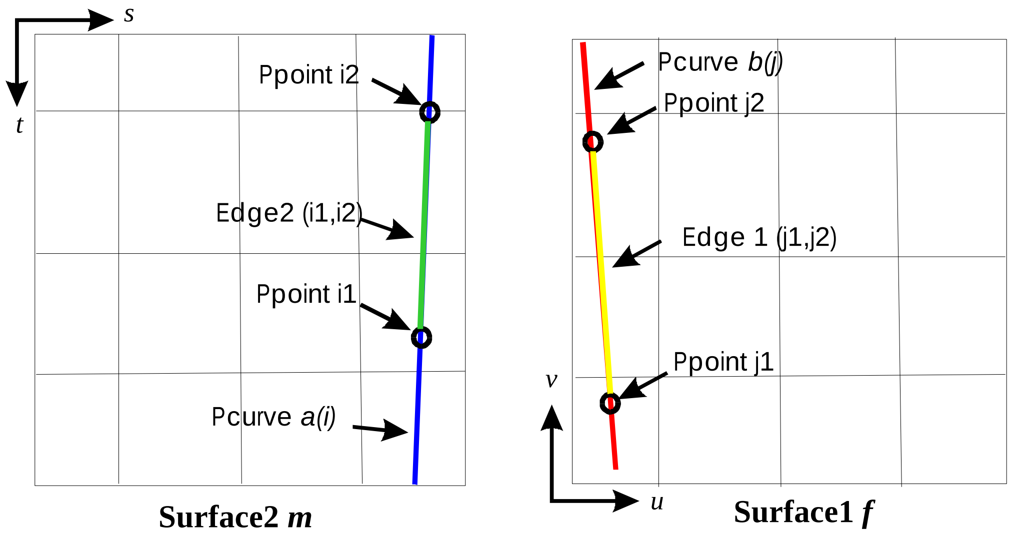

Figure 11 - CoEdge Domain Mapping

Here the *CoEdge* is defined as the set of *Edges* (“Edge1”, “Edge2”) and the *CoEdge*Spline :*g (r)*® *(i,j)* , where *Ppoint* i1 ≤ i ≤ i2 and *Ppoint* j1 ≤ j ≤ j2. *CoPoints* would exists as the set of physically coincident *Ppoints* (“PPI1”, “PPJ1”) and (“PPI2, “PPJ2”) and a *CartesianPoint*.

With this topology the ability to traverse boundaries, both logically, explicitly, and with information on the relationship of *Surface* parameter space affords many advantages. Clearly the ability to grid or mesh across trim *Surface* boundaries with node continuity is the most obvious.

The advantages inherent in GOBS can be summarized as the result of two factors. The first is the dropping of restrictions that geometry generation systems impose. In dropping these restrictions some new objects are required to allow the composition and traversal of a complex structure.

APPENDIX B – THREE BOX TEST CASE